The Polyphase Implementation of FIR Filters

Implementation of Digital Signal Processing

Sabih H. Gerez
University of Twente

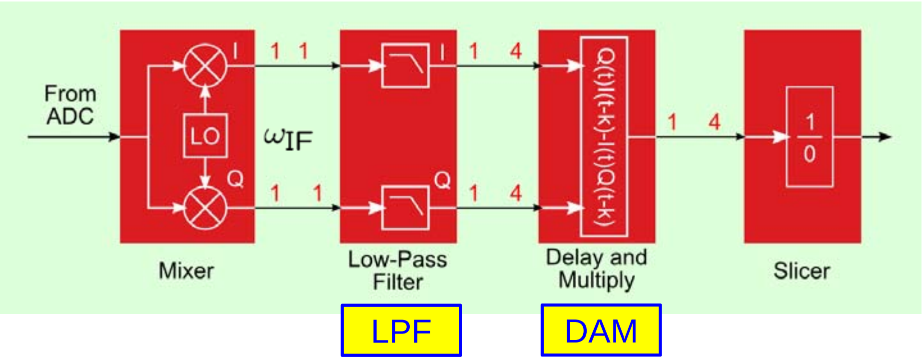
OUTLINE

- Downconversion and downsampling
- Polyphase implementation
- Upsampling

LITERATURE

- To study:
  - Langlois, J.M.P., D. Al-Khalili and R.J. Inkol, *Polyphase Filter Approach for High Performance, FPGA-Based Quadrature Demodulation*, Journal of VLSI Signal Processing, Vol.32, pp. 237-254, (2002).
- Optional texts for in-depth information:
  - Vaidyanathan, P.P., *Multirate Digital Filters, Filter Banks, Polyphase Networks, and Applications: A Tutorial*, Proceedings of the IEEE, Vol.78(1), pp. 56-93, (January 1990).
  - Harris, F.J., *Multirate Signal Processing for Communication Systems*, Prentice Hall, PTR, Upper Saddle River, New Jersey, (2004).

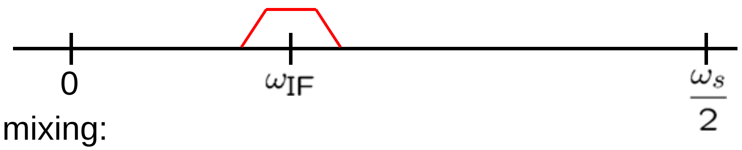
DEMODULATOR BLOCK DIAGRAM



The 16 samples per transmitted bit are first reduced to 4 and later back to 1.

SPECTRUM

- Before downconversion:

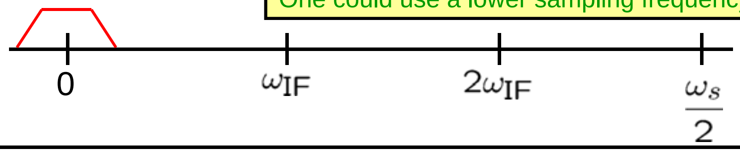


Sample frequency divided by 2

- After mixing:

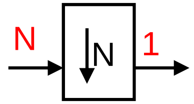
- After low-pass filtering:

One could use a lower sampling frequency at this stage

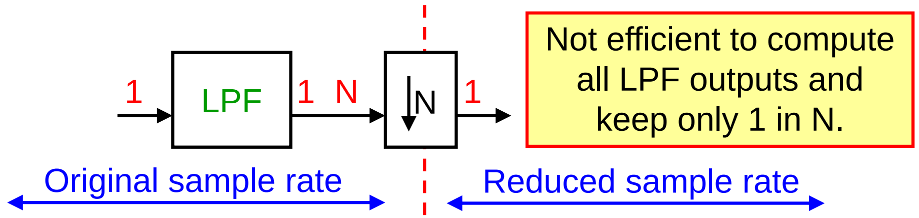


DOWNSAMPLING

- Operation in DSP where 1 out of N samples is kept (the other N-1 are thrown away).

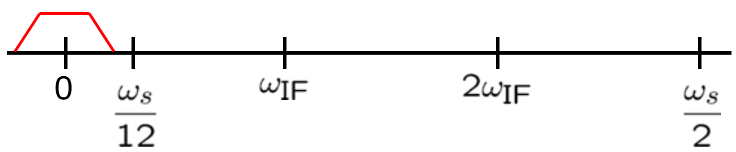


- Sometimes also called *decimation.*
- In example application, this would amount to:

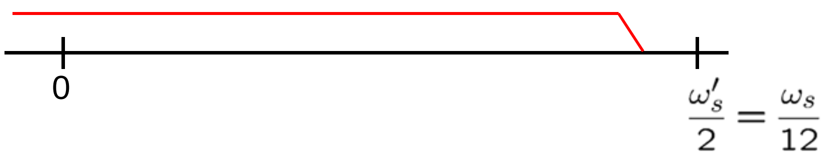


DOWNSAMPLING IN FREQUENCY DOMAIN

- Spectrum before downsampling:

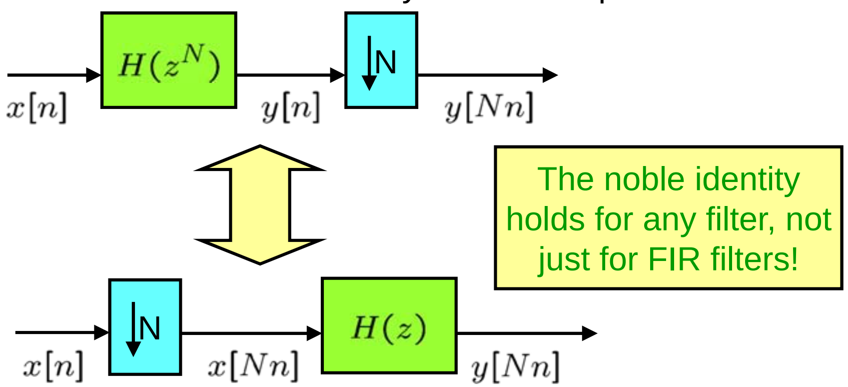


- Spectrum after downsampling with factor 6:



NOBLE IDENTITY FOR DOWNSAMPLING

- The context is a filter followed by a downsampler:



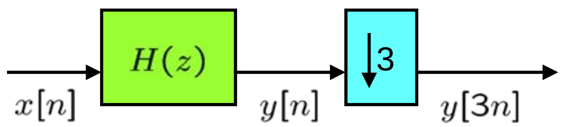
- *Interpretation:* filtering and downsampling can be swapped provided that delays in filter are N-fold (normally not true!).

POLYPHASE FILTERING EXAMPLE (1)

- Consider K[th]-order FIR filter with transfer function *H* given by coefficients *b*:

$$y[n] = \sum_{k=0}^{K} b[k] \cdot x[n-k]$$

- Example: downsampling by 3 after filtering, how to implement efficiently?



- Consider outputs after downsampling and rewrite by grouping coefficients with offsets of 3:

$$y[3n] = \sum_{k=0}^{K} b[k] \cdot x[3n-k]$$

$$= \sum_{k=0}^{K_0} b[3k] \cdot x[3(n-k)] + \sum_{k=0}^{K_1} b[3k+1] \cdot x[3(n-k)-1] + \sum_{k=0}^{K_2} b[3k+2] \cdot x[3(n-k)-2]$$

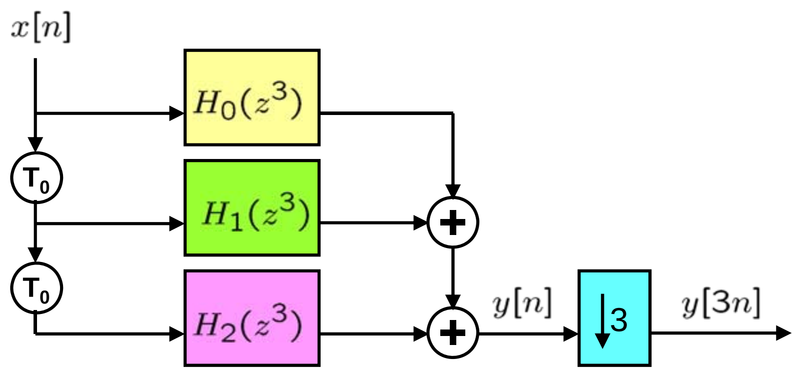
FIR filter with coefficients b[3k] applied to x[3n] $H_0(z^3)$

FIR filter with coefficients b[3k+1], applied to delayed x $H_1(z^3)$

FIR filter with coefficients b[3k+2], applied to x delayed twice $H_2(z^3)$

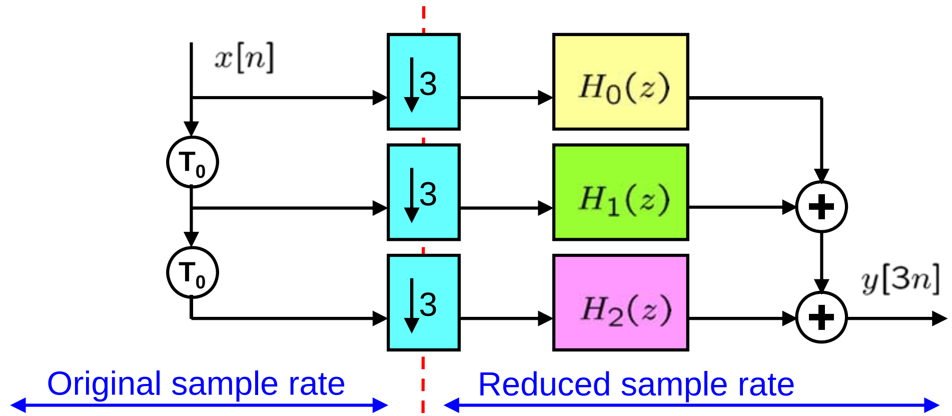
POLYPHASE FILTERING EXAMPLE (3)

- Graphical representation of rewriting:



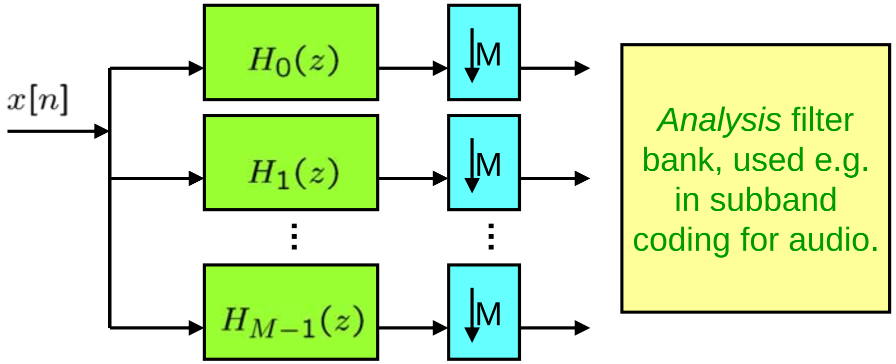
POLYPHASE FILTERING EXAMPLE (4)

- Now the noble identity can be applied to the three subfilters:



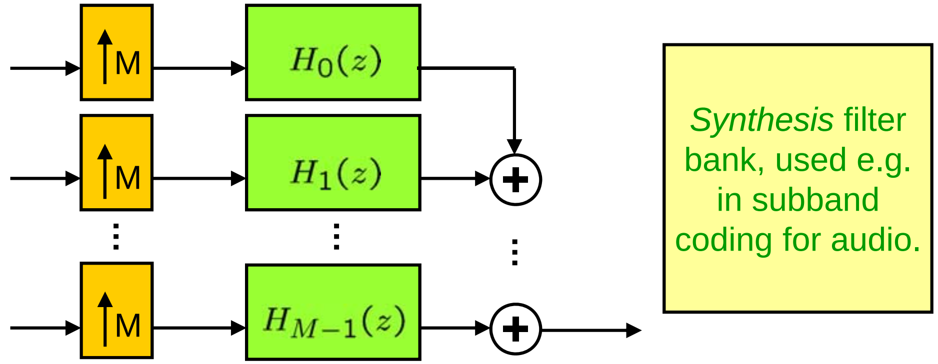
FILTER BANKS (1)

- Separate signal into adjacent frequency bands, by means of band-pass filters
- Each band has limited bandwidth and can therefore reduce its sample rate, polyphase solution can be applied!



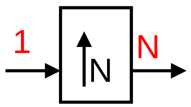
FILTER BANKS (2)

- Signal reconstruction after subband processing requires *upsampling.*
- Filtering after upsampling is required to avoid aliasing.



UPSAMPLING

- Operation in DSP where N samples are produced for each input (N-1 zeros are inserted between original samples)

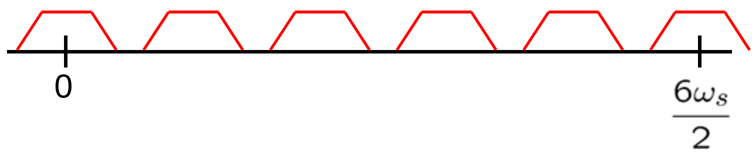


- Sometimes also called *interpolation.*

UPSAMPLING IN FREQUENCY DOMAIN

- Spectrum before upsampling:

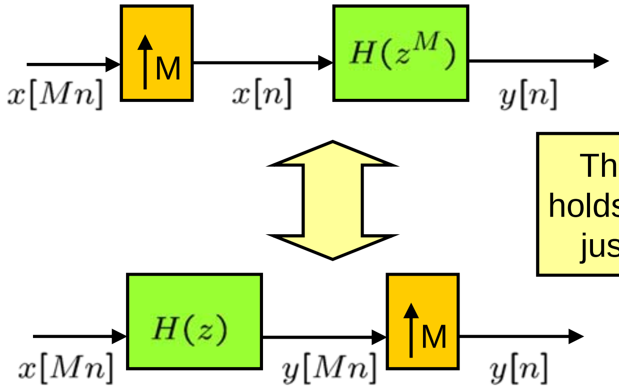
- Spectrum after upsampling with factor 6:



- Low-pass filtering is necessary to remove aliases.
- It is not efficient to feed zeros to filter.

THE NOBLE IDENT. FOR UPSAMPLING

- The context is an upsampler followed by a filter:



The noble identity holds for any filter, not just for FIR filters!

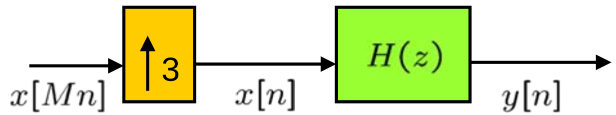
- *Interpretation:* filtering and upsampling can be swapped provided that delays in filter are M-fold (normally not true!).

POLYPHASE FILTERING EXAMPLE (1)

- Consider K[th]-order FIR filter with transfer function *H* given by coefficients *b*:

$$y[n] = \sum_{k=0}^{K} b[k] \cdot x[n-k]$$

- Example: upsampling by 3 followed by filtering, how to implement efficiently?



POLYPHASE FILTERING EXAMPLE (2)

- Start with definition, and group by coefficient index:

$$\begin{aligned} y[n] &= \sum_{k=0}^{K} b[k] \cdot x[n-k] \\ &= \sum_{k=0}^{K_0} b[3k] \cdot x[n-3k] + \\ &\quad \sum_{k=0}^{K_1} b[3k+1] \cdot x[n-3k-1] + \\ &\quad \sum_{k=0}^{K_2} b[3k+2] \cdot x[n-3k-2] \end{aligned}$$

Depending on *n*, only one out of three groups will be unequal to zero!

POLYPHASE FILTERING EXAMPLE (3)

- Now consider outputs with different offsets separately and keep only those inputs unequal to zero.
- The result consists of three sequences that are filtered versions of the signal before upsampling.

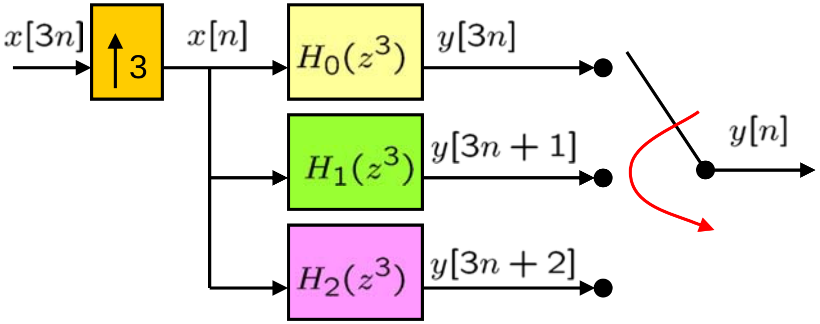
$$y[3n] = \sum_{k=0}^{K_0} b[3k] \cdot x[3(n-k)] \qquad \boxed{H_0(z^3)}$$

$$y[3n+1] = \sum_{k=0}^{K_1} b[3k+1] \cdot x[3(n-k)] \qquad \boxed{H_1(z^3)}$$

$$y[3n+2] = \sum_{k=0}^{K_2} b[3k+2] \cdot x[3(n-k)] \qquad \boxed{H_2(z^3)}$$

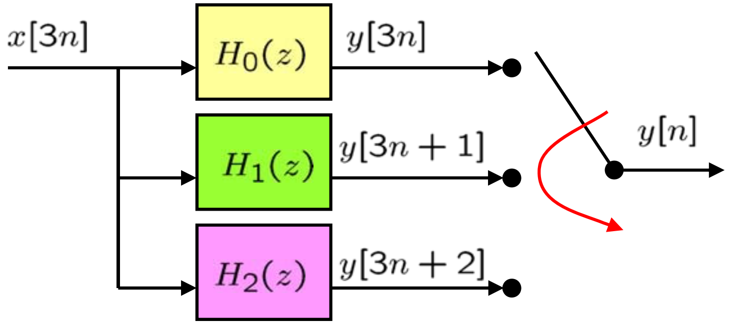
POLYPHASE FILTERING EXAMPLE (4)

- The previous equations represent:



POLYPHASE FILTERING EXAMPLE (5)

- After applying the noble identity for upsampling:



- *Note:* the upsample nodes have been left out as they produce zeros when the switch is not using their outputs.